

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-112750

(43)Date of publication of application : 21.04.2000

(51)Int.Cl.

G06F 9/06

(21)Application number : 10-296188

(71)Applicant : HYPER TEC:KK

(22)Date of filing :

02.10.1998

(72)Inventor : OGAWA MUTSUMI

(54) INFORMATION PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To make copying an illegal program very difficult by having a subprogram that has pointer information comprising specific storage medium identifiers.

SOLUTION: This processor has a sub-module performing a part of information processing and a subprogram having pointer information comprising identifiers for identifying a storage medium where other subprograms are stored. When a start designation instruction is inputted from an inputting part 101, an information processing part 105 instructs a program reading part 104 to read a storage medium identifier belonging to the start designation



instruction and a subprogram identified by an address. The part 105 sets a program counter (P) in the subprogram to one, decides if the P-th instruction exists in the subprogram and whether or not the P-th instruction is a jump instruction to the next subprogram having pointer information consisting of a storage medium identifier and an address and interprets the P-th instruction and performs processing if it is not the jump instruction.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-112750

(P2000-112750A)

(43) 公開日 平成12年4月21日 (2000.4.21)

(51) Int.Cl.⁷

G 0 6 F 9/06

識別記号

5 5 0

F I

G 0 6 F 9/06

テーマコード (参考)

5 5 0 Z 5 B 0 7 6

審査請求 未請求 請求項の数6 F D (全 12 頁)

(21) 出願番号 特願平10-296188

(22) 出願日 平成10年10月2日 (1998.10.2)

(71) 出願人 398055691

株式会社ハイパーテック

京都府京都市下京区中堂寺南町17 京都リ

サーチパーク (財) 京都高度技術研究所

内

(72) 発明者 小川 睦美

京都市左京区岩倉花園町7番地の3

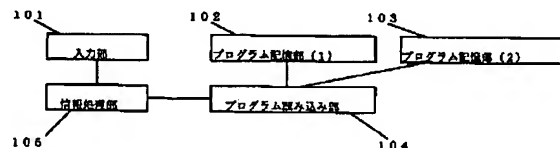
Fターム (参考) 5B076 FA00

(54) 【発明の名称】 情報処理装置

(57) 【要約】

【課題】 本発明は、プログラムやデータの複製を作成する場合に、プログラム等のポインタ情報に記憶媒体識別子を入れることにより、他の記憶媒体への不正な複製をさせないことを目的とする。

【解決手段】 一連の情報処理を行うプログラムを複数のサブプログラムに分割し、複数の記憶媒体に分割して記憶されることを特徴とする情報処理装置であって、記憶媒体を識別するための記憶媒体識別子からなるポインタ情報を有することを特徴とする情報処理装置。



【特許請求の範囲】

【請求項 1】 一連の情報処理を行うプログラムを複数のサブプログラムに分割し、前記複数のサブプログラムが複数の記憶媒体に分割して記憶されるプログラム構成方法により構成されている情報処理装置であって、前記サブプログラムが情報処理の一部を行うサブモジュールと他のサブプログラムが記憶されている記憶媒体を識別するための記憶媒体識別子からなるポインタ情報を有することを特徴とする情報処理装置。

【請求項 2】 ポインタ情報が他のサブプログラムが記憶されている記憶媒体を識別するための記憶媒体識別子と当該記憶媒体識別子で識別される記憶媒体中のアドレスを識別するアドレス識別子からなる請求項 1 記載の情報処理装置。

【請求項 3】 一連の情報処理を行うプログラムを複数のサブプログラムに分割するプログラム構成方法により構成されている情報処理装置であって、前記サブプログラムが情報処理の一部を行うサブモジュールと他のサブプログラムが記憶されているアドレスを識別するアドレス識別子からなるポインタ情報を有し、当該アドレス識別子をサブプログラム中の予め決められたプログラム変数の値をパラメータとする関数から生成することを特徴とする情報処理装置。

【請求項 4】 記憶媒体識別子をサブプログラム中の予め決められたプログラム変数の値をパラメータとする関数から生成することを特徴とする請求項 1 または請求項 2 記載の情報処理装置。

【請求項 5】 アドレス識別子をサブプログラム中の予め決められたプログラム変数の値をパラメータとする関数から生成することを特徴とする請求項 2 記載の情報処理装置。

【請求項 6】 アドレス識別子と記憶媒体識別子の双方をサブプログラム中の予め決められた異なるプログラム変数の値をパラメータとする関数から生成することを特徴とする請求項 2 記載の情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 コンピュータソフトウェアの不正コピーを防止または困難にするための構成を有するプログラム構成方法により構成された情報処理装置に関するものである。

【0002】

【従来の技術】 従来の技術として、特開平 2-21344 がある。本発明は、ファイルのコピーを防止するためにファイルを分割しランダムな空き領域に書き込み、ファイルを復元する情報はファイル・アロケーション・テーブルという表で一括管理するものである。

【0003】

【発明が解決しようとする課題】 しかしながら、従来の技術では、ファイルを復元する情報が一括管理されてお

り、復元するための情報を一箇所見つけることができれば、容易に復元可能、つまりファイルのコピーが可能となる、という課題を有していた。また、一括管理された情報が復元するための情報であることが比較的容易に判断できた。

【0004】

【課題を解決するための手段】 請求項 1 の発明は、一連の情報処理を行う複数のプログラムであり、情報処理の一部を行うサブモジュールと他のサブプログラムが記憶されている記憶媒体を識別するための記憶媒体識別子からなるポインタ情報を有するサブプログラムを有することを特徴としている。

【0005】 請求項 2 の発明は、請求項 1 の発明において、ポインタ情報が他のサブプログラムが記憶されている記憶媒体を識別するための記憶媒体識別子とその記憶媒体識別子で識別される記憶媒体中のアドレスを識別するアドレス識別子からなることを特徴としている。請求項 3 の発明は、請求項 1 の発明において、一連の情報処理を行う複数のプログラムであり、情報処理の一部を行うサブモジュールと他のサブプログラムが記憶されているアドレスを識別するアドレス識別子からなるポインタ情報を有し、そのアドレス識別子をサブプログラム中の予め決められたプログラム変数の値をパラメータとする関数から生成するサブプログラムを有することを特徴としている。

【0006】 請求項 4 の発明は、請求項 2 の発明において、記憶媒体識別子をサブプログラム中の予め決められたプログラム変数の値をパラメータとする関数から生成することを特徴としている。請求項 5 の発明は、請求項 2 の発明において、アドレス識別子をサブプログラム中の予め決められたプログラム変数の値をパラメータとする関数から生成することを特徴としている。

【0007】 請求項 6 の発明は、請求項 2 の発明において、アドレス識別子と記憶媒体識別子の双方をサブプログラム中の予め決められた異なるプログラム変数の値をパラメータとする関数から生成することを特徴としている。

【0008】

【発明の実施の形態】 以下、本発明の実施の形態を図面を用いて詳細に説明する。

実施の形態 1 図 1 は本発明に係る情報処理装置のブロック図で、入力部 101 と、プログラム記憶部 (1) 102 と、プログラム記憶部 (2) 103 と、プログラム読み込み部 104 と、情報処理部 105 とからなる。

【0009】 入力部 101 は、例えば、キーボードやマウスなどで、情報処理の開始指示および当該情報処理の命令群の開始アドレスを特定する記憶媒体識別子および記憶媒体識別子で識別される記憶媒体中のアドレスからなる開始指示命令を入力する。プログラム記憶部 (1) 102 は、例えば、ハードディスクやフロッピーディス

クなどであり、一連の情報処理の命令群をサブプログラムと呼ぶ一部の情報処理の命令群を有する情報に分割して記憶している。そして、当該サブプログラムは、情報処理の一部を行うための命令の集合であるサブモジュールと次の情報処理の一部を行うサブプログラムを識別するポインタ情報を有する。

【0010】プログラム記憶部(2)103は、プログラム記憶部(1)102とは別の記憶媒体であり、CD-ROMやプロテクトモジュールなどである。ここで、プロテクトモジュールとは、プリンタのコネクタに接続する小容量のメモリを持った記憶媒体で、プロテクトモジュールに記憶されている情報を読み込むには、特別なドライバが必要となる。

【0011】プログラム読み込み部104は、指示に従ってプログラム記憶部(1)102またはプログラム記憶部(2)103に記憶されているサブプログラムを読み込む。このプログラム読み込み部104は、プログラム記憶部(1)102とプログラム記憶部(2)103に記憶されているサブプログラムを読み込むためのドライバを有し、サブプログラムの読み込みの指示に含まれる記憶媒体識別子で識別される記憶媒体用のドライバを起動して、サブプログラムを読み込む。

【0012】情報処理部105は、例えばコンピュータの演算処理部(CPU)などがあり、入力部101から入力された開始指示命令が有する記憶媒体識別子とアドレスで特定されるサブプログラムの読み込みをプログラム読み込み部104に指示し、またプログラム読み込み部104が読み込んだサブプログラムを処理する。次に図1に示す情報処理装置における情報処理の動作を図2のフローチャートを参照しながら説明する。

【0013】ステップS201 情報処理部105は、入力部101から開始指示命令が入力されたか否か判断する。開始指示命令が入力されれば、ステップS202に行き、入力されていない場合はステップS201に戻る。

ステップS202 情報処理部105は、開始指示命令が有する記憶媒体識別子とアドレスで識別されるサブプログラムを読み込むことをプログラム読み込み部104に指示する。

【0014】ステップS203 プログラム読み込み部104はステップS202で識別されたサブプログラムを読み込む。

ステップS204 情報処理部105は、サブプログラム中のプログラムカウンタ(P)を1にセットする。

ステップS205 情報処理部105は、P番目の命令がサブプログラムに存在するか否か判断する。P番目の命令が存在すれば、ステップS206に行き、存在しなければ処理を終了する。

【0015】ステップS206 情報処理部105は、P番目の命令が記憶媒体識別子とアドレスからなるポ

インタ情報を有する次のサブプログラムへのジャンプ命令か否か判断する。

次のサブプログラムへのジャンプ命令であればステップS202に戻る。当該ジャンプ命令でなければステップS207に行く。

【0016】ステップS207 情報処理部105は、P番目の命令を解釈し、処理する。

ステップS208 情報処理部105は、PにP+1を代入する。ステップS205に戻る。

次に上記情報処理装置において、プログラム記憶部

(1)102がハードディスクであり、プログラム記憶部(2)103がCD-ROMである場合に、複数のサブプログラムがハードディスク(プログラム記憶部(1)102)とCD-ROM(プログラム記憶部(2)103)に分割されている場合の具体的な動作について説明する。

【0017】今、図3に示されるようにハードディスク(プログラム記憶部(1)102)とCD-ROM(プログラム記憶部(2)103)に一連の情報処理の一部を行うサブプログラムが分割して記憶されている。そして、各サブプログラムは、次のサブプログラムへ処理を移行するための記憶媒体識別子とアドレスからなるポインタ情報が存在する。

【0018】かかる場合に、入力部101から「処理開始」「ハードディスク」「0」という開始指示命令が入力されるとする。ここで、「処理開始」「ハードディスク」「0」の「処理開始」は命令であり、「ハードディスク」は記憶媒体識別子であり、「0」はアドレスである。そして、「ハードディスク」と「0」で最初に処理するサブプログラムが「サブプログラム1」と決定される。

【0019】上記の開始指示命令が入力されれば、情報処理部105は、ハードディスク上のサブプログラム1を読み込むようにプログラム読み込み部104に指示し、プログラム読み込み部104はサブプログラム1を読み込む。そして、情報処理部105は順次サブプログラム1中の命令を順次処理し、「JAMP」「CD-ROM」「BBB」の命令の処理によりCD-ROM上のサブプログラム2に処理は移行する。

【0020】なお、「JAMP」「CD-ROM」「BBB」の命令の以降に他の命令が存在するが、その命令の存在により、次のサブプログラムへの処理への移行命令であることが解析しにくくなっている。また、「JAMP」「CD-ROM」「BBB」の命令では、記憶媒体識別子は「CD-ROM」となっているが、説明の都合上記憶媒体名を記載しているが、実際のプログラムでは「CD-ROM」であることを識別するための数値などの識別子でも良い。

【0021】サブプログラム2に処理が移行するとは、情報処理部105が、プログラム読み込み部104にC

D-ROM上のサブプログラム2を読み込むことを指示し、上記の説明と同様にプログラム読み込み部104はサブプログラム2を読み込み、情報処理部105は順次サブプログラム2中の命令を順次処理することを意味する。

【0022】そのサブプログラム2の命令の処理中の「JAMP 'ハードディスク' '456'」の命令の処理によりハードディスク上のサブプログラム3に処理は移行する。以降、同様にサブプログラム4からサブプログラム8が処理されて、一連の処理が終了する。

【0023】なお、上記の具体的な動作の説明において、プログラム記憶部(1)102がハードディスクであり、プログラム記憶部(2)103がCD-ROMであったが、それぞれのプログラム記憶部がフロッピーディスクやプロテクトモジュールなどの記憶媒体であっても良いし、他方のプログラム記憶部がネットワーク上の他のマシンのハードディスクなどの記憶媒体であってもよい。他方のプログラム記憶媒体がネットワーク上の他のマシンのハードディスクである場合とは、例えば、インターネット上のURLで識別されるホームページであり、当該ホームページからHTTPなどのプロトコルによりサブプログラムを逐次取り出して、一連の処理を行ってもよい。

【0024】また、上記の具体的な動作の説明において、2つの異なる記憶媒体であるプログラム記憶部にサブプログラムを分割して持たせたが、3つ以上の異なる記憶媒体にサブプログラムを分割してもよい。また、図4に示すように、他のサブプログラムへのポインタ情報が記憶媒体識別子だけでもよい。図4のようにプログラムがサブプログラムに分割された場合は、入力部101から「処理開始 アドレス識別子」(この場合「処理開始 '0'」)が入力され、デフォルトの記憶媒体であるハードディスクの「アドレス識別子 '0'」で示されたプログラムを処理する。従って、情報処理部105は、ハードディスク上のサブプログラム1をまず処理し、次に記憶媒体識別子(CD-ROM)で識別される記憶媒体中の予め決められたアドレス(0)に記憶されたサブプログラム(サブプログラム2)を処理し、続いてデフォルトの記憶媒体であるハードディスク上の「処理開始 アドレス識別子」の後のサブプログラム(サブプログラム3)を処理することとなる。

【0025】この場合、CD-ROMの中のプログラムはアドレス'0'から開始されていることが前提となっている。また、次に移行するサブプログラムを特定するポインタ情報が記憶媒体識別子とアドレスであったが、記憶媒体の種類によりトラック番号やセクタ番号などの異なるものであってもよい。

【0026】以上のように本実施の形態で説明したプログラム構成方法によりプログラムを構成すれば、単にコピーして、分割されたプログラムを合成しても、記憶媒

体識別子がプログラム中に存在することにより、当該記憶媒体識別子で識別される記憶媒体を読みに行くことになり、正常に一連の情報処理が行われず、コピーの効果を発揮できない。従って、不正コピーを有効に防止することができる。

【0027】さらに、例えば、本情報処理装置の処理はデータの検索処理であり、「Jump CD-ROM」の後ろに「If (error) ……」があり、CD-ROMへのジャンプ命令のエラー処理が記載されている場合(図5参照)、エラーの時は遅い検索(例えば、逐次検索)を行い、正常な時(CD-ROMのサブプログラム2の処理)は速い検索(例えば、二分検索)を行うこととする、ということも可能である。かかる場合、不正コピーした情報処理装置は、正常な情報処理装置と比較して非常に遅い情報処理(検索)しかできない、ということになり、不正コピー者に対して正規なサービスを受けないという不具合を与えることができる。

【0028】実施の形態2 図6は本発明に係る情報処理装置のブロック図で、入力部101と、プログラム記憶部602と、プログラム読み込み部104と、情報処理部605とからなる。プログラム記憶部602は、例えば、ハードディスクやフロッピーディスクなどであり、一連の情報処理の命令群をサブプログラムと呼ぶ一部の情報処理の命令群を有する情報に分割して記憶している。そして、当該サブプログラムは、情報処理の一部を行うための命令の集合であるサブモジュールと次の情報処理の一部を行うサブシステムの開始位置を示すアドレスを有する。

【0029】そして、アドレスは、前記サブプログラム中の一部のプログラムコードとして使用されている予め決められたプログラム変数の値をパラメータとした関数により生成されることとする。情報処理部605は、入力部101から入力された開始指示命令が有するアドレスで特定されるサブプログラムの読み込みをプログラム読み込み部104に指示し、またプログラム読み込み部104が読み込んだサブプログラムを処理する。

【0030】次に、上記図6に示す情報処理装置における情報処理の動作を図7のフローチャートを参照しながら説明する。

ステップS701 情報処理部605は、入力部101から開始指示命令が入力されたか否かを判断する。開始指示命令が入力されれば、ステップS702に行き、入力されていないならばステップS701に戻る。

【0031】ステップS702 情報処理部605は、開始指示命令が有するアドレスで識別されるサブプログラムを読み込むことをプログラム読み込み部104に指示する。

ステップS703 プログラム読み込み部104はステップS702で識別されたサブプログラムを読み込む。

【0032】ステップS704 情報処理部605は、

サブプログラム中のプログラムカウンタ (P) を1にセットする。

ステップS705 情報処理部605は、P番目の命令がサブプログラムに存在するか否か判断する。P番目の命令が存在すれば、ステップS706に行き、存在しなければ処理を終了する。

【0033】ステップS706 情報処理部605は、P番目の命令が予め決められたプログラム変数をパラメータとした関数からなるポインタ情報を有する次のサブプログラムへの移行命令が否か判断する。次のサブプログラムへの移行命令であればステップS707に行き、当該移行命令でなければステップS708に行く。

【0034】ステップS707 情報処理部605は、プログラム変数の値を読みだし、アドレスを算出するための予め決められた関数を適用してアドレスを算出する。ステップS702に戻る。

ステップS708 情報処理部605は、P番目の命令を解釈し、処理する。

ステップS709 情報処理部605は、PにP+1を代入する。ステップS705に戻る。

【0035】次に、上記情報処理装置において、プログラム記憶部602に図8のような分割されたサブプログラムが複数記憶されている場合の具体的な動作について説明する。今、図8に示されるように、例えばハードディスク（プログラム記憶部602）にサブプログラムが分割して記憶されている。

【0036】かかる場合に、入力部101から「処理開始 '123' 」という開始指示命令が入力されるとする。ここで、「処理開始 '123' 」の「処理開始」は命令であり、「'123' 」はアドレスである。そして、「'123' 」のアドレスにより最初に処理するサブプログラムがサブプログラム1と決定される。上記の開始指示命令が入力されれば、情報処理部605は、プログラム記憶部602のアドレス '123' で始まるサブプログラム1を読み込むようにプログラム読み込み部104に指示し、プログラム読み込み部104はサブプログラム1を読み込む。そして、情報処理部605は順次サブプログラム1中の命令を順次処理し、予め決められた変数「A」をパラメータとする予め決められた関数「f1」の適用により算出された値を次に処理するサブプログラムの開始アドレスだと判断し、つまり「f1 (A) 」の命令の処理により算出された値を次に処理するサブプログラムの開始アドレスであると判断し、プログラム読み込み部104に当該算出したアドレスで開始されるサブプログラムの読み込みを指示する。

【0037】今、Aの値が「4」であり「f1 (x) = 111x + 12」だとすると、f1 (A) の計算により「'456' 」が算出され、次にサブプログラム2に処理は移行する。そして、情報処理部605は順次サブプログラム2中の命令を順次処理し、予め決められた変数

「B」をパラメータとする関数の適用により算出された値を次に処理するサブプログラムの開始アドレスだと判断し、つまり「f2 (B) 」の命令の処理により算出された値を次に処理するサブプログラムの開始アドレスであると判断し、プログラム読み込み部104に当該算出したアドレスで開始されるサブプログラムの読み込みを指示する。

【0038】今、例えばBの値が「2」であり、「f2 (x) = 2x - 4」だとすると、f2 (B) の処理により「'0' 」が算出され、次にサブプログラム3に処理は移行する。情報処理部605は順次サブプログラム3中の命令を順次処理し、サブプログラム3中には予め決められた次のサブプログラムのアドレスを算出するためのプログラム変数が存在しなかったために、サブプログラム3の全処理を完了して、一連の情報処理は終了する。

【0039】なお、上記の具体的な動作の説明において、アドレスを算出するための関数が予め決められた関数としたが、サブプログラム中に埋め込まれて他の処理で使用する関数であっても良い。かかる他の処理で使用する関数によりアドレスを算出するほうが、不正コピーのための命令であることが解析されにくい。プログラム変数も同様に、他の処理で使用される変数でも良いし、アドレス算出用の変数でも良い。

【0040】以上のように本実施の形態で説明したプログラム構成方法によりプログラムを構成すれば、サブプログラム中で使用されている予め決められた変数をパラメータとする関数を適用してアドレスを算出し、当該アドレスにより次のサブプログラムを決定するので、たとえ分割されたサブプログラムをすべて読み出すことができたとしても、どのように連結するのかを判断することが極めて困難であり、不正コピーは非常に困難となる。

【0041】実施の形態3 図9は本発明に係る情報処理装置のブロック図で、入力部101と、プログラム記憶部(3)902と、プログラム記憶部(4)903と、プログラム読み込み部104と、情報処理部905とからなる。プログラム記憶部(3)902は、例えば、ハードディスクやフロッピーディスクなどであり、一連の情報処理の命令群をサブプログラムと呼ぶ一部の情報処理の命令群を有する情報に分割して記憶している。そして、当該サブプログラムは、情報処理の一部を行うための命令の集合であるサブモジュールと次の情報処理の一部を行うサブプログラムを識別する記憶媒体識別子と記憶媒体識別子で識別される記憶媒体中の位置を示すアドレスを有する。

【0042】そして、記憶媒体識別子とアドレスは、前記サブプログラム中の一部のプログラムコードとして使用されている予め決められたプログラム変数の値をパラメータとした関数により各々生成されることとする。プログラム記憶部(4)903は、プログラム記憶部

(3) 902とは別の記憶媒体であり、CD-ROMやプロテクトモジュールなどであり、記憶している内容および構成はプログラム記憶部(3) 902と同様に、サブプログラムを1つ以上記憶している。

【0043】情報処理部905は、入力部101から入力された開始指示命令が有する記憶媒体識別子とアドレスで特定されるサブプログラムの読み込みをプログラム読み込み部104に指示し、またプログラム読み込み部104が読み込んだサブプログラムを処理する。次に図9に示す情報処理装置における情報処理の動作を図10のフローチャートを参照しながら説明する。

【0044】ステップS1001 情報処理部905は、入力部101から開始指示命令が入力されたか否かを判断する。開始指示命令が入力されれば、ステップS1002に行き、入力されていないければステップS1001に戻る。

ステップS1002 情報処理部905は、開始指示命令が有する記憶媒体識別子とアドレスで識別されるサブプログラムを読み込むことをプログラム読み込み部104に指示する。

【0045】ステップS1003 プログラム読み込み部104はステップS1002で識別されたサブプログラムを読み込む。

ステップS1004 情報処理部905は、サブプログラム中のプログラムカウンタ(P)を1にセットする。ステップS1005 情報処理部905は、P番目の命令がサブプログラムに存在するか否かを判断する。P番目の命令が存在すれば、ステップS1006に行き、存在しなければ処理を終了する。

【0046】ステップS1006 情報処理部905は、P番目の命令が予め決められたプログラム変数をパラメータとした関数からなるポインタ情報を有する次のサブプログラムへの移行命令が否かを判断する。次のサブプログラムへの移行命令であればステップS1007に行き、当該移行命令でなければステップS1008に行く。

【0047】ステップS1007 情報処理部905は、プログラム変数の値を読み出し、関数を適用して記憶媒体識別子およびアドレスを算出する。ステップS1002に戻る。

ステップS1008 情報処理部905は、P番目の命令を解釈し、処理する。

【0048】ステップS1009 情報処理部905は、PにP+1を代入する。ステップS1005に戻る。

次に上記情報処理装置において、プログラム記憶部

(3) 902がハードディスクであり、プログラム記憶部(4) 903がプロテクトモジュールである場合に、複数のサブプログラムがハードディスク(プログラム記憶部(3) 902)とプロテクトモジュール(プログラ

ム記憶部(4) 903)に分割されている場合の具体的な動作について説明する。

【0049】今、図11に示されるようにハードディスク(プログラム記憶部(3) 902)とプロテクトモジュール(プログラム記憶部(4) 903)に一連の情報処理の一部を行うサブプログラムが分割して記憶されている。本具体例において、A1、A2、A3、A4は記憶媒体識別子を算出するためのプログラム変数であり、B1、B2、B3、B4はアドレスを算出するためのプログラム変数であり、hは記憶媒体識別子を算出する関数であり、fはアドレスを算出する関数である。

【0050】かかる場合に、入力部101から「処理開始「ハードディスク」「345」という開始指示命令が入力されるとする。この開始指示命令が入力されれば、情報処理部905は、ハードディスク上のサブプログラム1を読み込むようにプログラム読み込み部104に指示し、プログラム読み込み部104はサブプログラム1を読み込む。そして、情報処理部905は順次サブプログラム1中の命令を順次処理し、予め決められたプログラム変数A1、B1に対する関数h、fを適用する命令が存在すれば、そのときのA1、B1の値にh、fの関数を適用して、記憶媒体識別子とアドレスを算出する。

【0051】具体的には、サブプログラム中の「h(A1)」の処理により「h(A)」となり、「h」を4つ先のアルファベットを出力する関数(入力が「A」なら出力は「E」、入力が「Y」なら出力が「C」となる)だとすれば「h(A)=E」となる。また、サブプログラム中の「f(B1)」の処理により「f(3)」となり、「f(x)=x-3」とすれば「f(3)=0」となる。

【0052】ここで、デバイスEはプロテクトモジュールだとすると、次にプロテクトモジュールのアドレス「0」で識別されるサブプログラムに処理が移行することとなる。そして、サブプログラム2を同様に処理する。続いて、情報処理部905は、サブプログラム2中の予め決められたプログラム変数A2、B2に対する関数h、fを適用する命令の存在を認知し、次に移行するサブプログラムを特定する記憶媒体識別子とアドレスを同様に算出する。

【0053】そして、サブプログラム3を処理する。以上の手順に従って、情報処理部905は、さらにサブプログラム4と5を処理し、一連の処理を終了する。なお、上記の具体的な動作の説明において、プログラム記憶部(3) 902がハードディスクであり、プログラム記憶部(4) 903がプロテクトモジュールであったが、それぞれのプログラム記憶部がフロッピーディスクやCD-ROMなどの記憶媒体であっても良いし、他方のプログラム記憶部がネットワーク上の他のマシンのハードディスクなどの記憶媒体であってもよい。他方のプ

ログラム記憶部がネットワーク上の他のマシンのハードディスクである場合とは、例えば、インターネット上のURLで識別されるホームページであり、当該ホームページからHTTPなどのプロトコルによりサブプログラムを逐次取り出して、一連の処理を行ってもよい。

【0054】また、上記の具体的な動作の説明において、2つの異なる記憶媒体であるプログラム記憶部にサブプログラムを分割して持たせたが、3つ以上の異なる記憶媒体にサブプログラムを分割してもよい。また、次に移行するサブプログラムを特定するポインタ情報が記憶媒体識別子とアドレスであったが、記憶媒体の種類によりトラック番号やセクタ番号などの異なるものであってもよい。

【0055】また、図4に示すように、他のサブプログラムへのポインタ情報が記憶媒体識別子だけでも良い。以上のように本実施の形態で説明した2つ以上の記憶媒体で分割してプログラムを保持するプログラム構成方法によりプログラムを構成すれば、単にコピーして、分割されたプログラムを合成しても、記憶媒体識別子がプログラム中に存在することにより、当該記憶媒体識別子で識別される記憶媒体を読みに行くことになり、正常に一連の情報処理が行われず、コピーの効果を発揮できない。従って、不正コピーを有効に防止することができる。

【0056】さらに、サブプログラム中で使用されている予め決められた変数をパラメータとする関数を適用してアドレスを算出し、当該アドレスにより次のサブプログラムを決定するので、たとえ分割されたサブプログラムを複数の記憶媒体から読んでいるがごとくプログラムコードを変更できたとしても、通常のプログラムでも使用されているプログラム変数をパラメータとして関数を適用して記憶媒体識別子およびアドレスを決定しているため、どのようにサブプログラムを連結するのかを解析することが極めて困難であり、また、どの命令で記憶媒体識別子とアドレスを決定しているかの解析が極めて困難となり、不正コピーが防止できる。

【0057】

【発明の効果】以上説明したように本発明によれば、分割されたサブプログラム中に各サブプログラムの連結情報（ポインタ情報）を分散させるとともに、ポインタ情報

に記憶媒体識別子を入れることにより、他の記憶媒体へのコピー後は動作しないこととなり、しかも、ポインタ情報中の記憶媒体識別子やアドレスなどをプログラムで使用しているプログラム変数をパラメータとする関数から算出するので、ポインタ情報がサブプログラム中のどこにあるのかを解析することが非常に困難になり、不正なプログラムのコピーが非常に困難となる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態における情報処理装置のブロック図である。

【図2】本発明の第1の実施の形態における情報処理装置の動作について説明するフローチャートである。

【図3】本発明の第1の実施の形態における複数のサブシステムに分割されたプログラム構成方法の例である。

【図4】本発明の第1の実施の形態における複数のサブシステムに分割されたプログラム構成方法の例である。

【図5】本発明の第1の実施の形態における複数のサブシステムに分割されたプログラム構成方法の例である。

【図6】本発明の第2の実施の形態における情報処理装置のブロック図である。

【図7】本発明の第2の実施の形態における情報処理装置の動作について説明するフローチャートである。

【図8】本発明の第2の実施の形態における複数のサブシステムに分割されたプログラム構成方法である。

【図9】本発明の第3の実施の形態における情報処理装置のブロック図である。

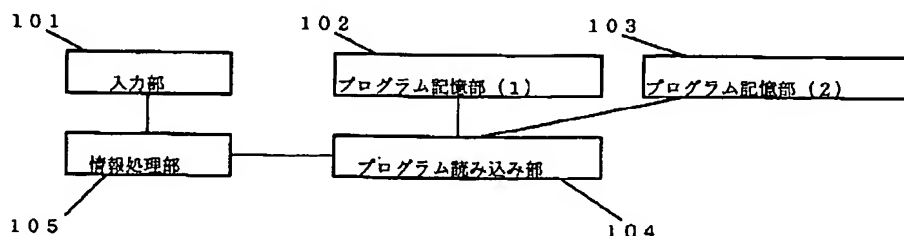
【図10】本発明の第3の実施の形態における情報処理装置の動作について説明するフローチャートである。

【図11】本発明の第3の実施の形態における複数のサブシステムに分割されたプログラム構成方法である。

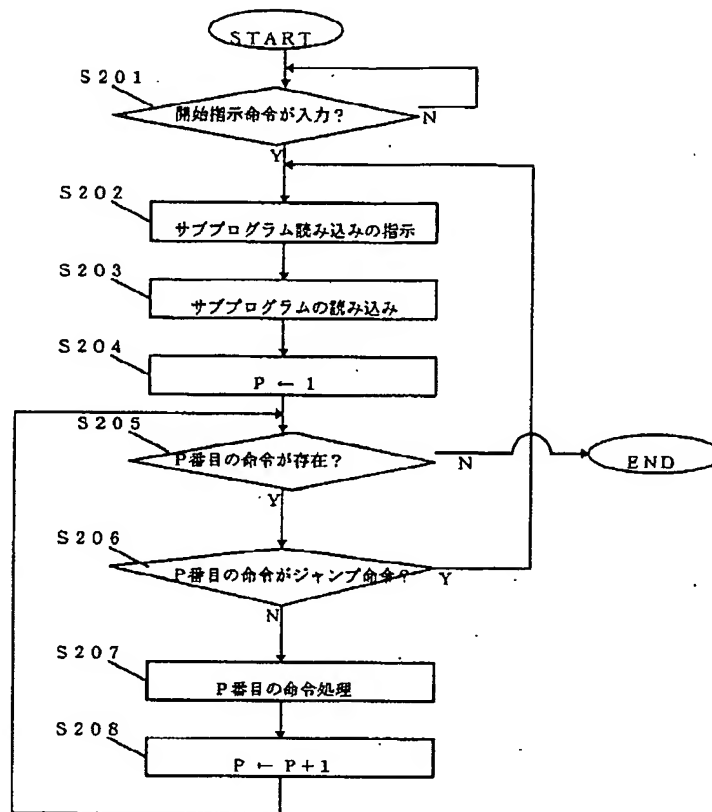
【符号の説明】

101 入力部
102 プログラム記憶部(1)
103 プログラム記憶部(2)
104 プログラム読み込み部
105, 605, 905 情報処理部
602 プログラム記憶部
902 プログラム記憶部(3)
903 プログラム記憶部(4)

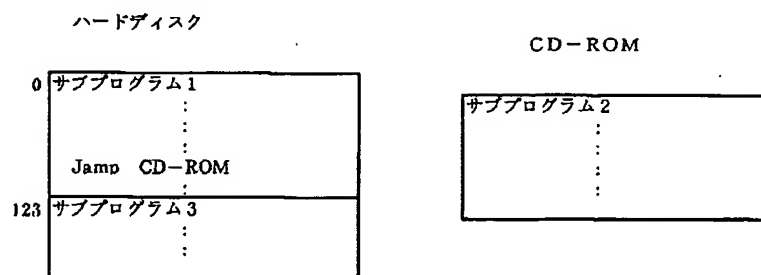
【図1】



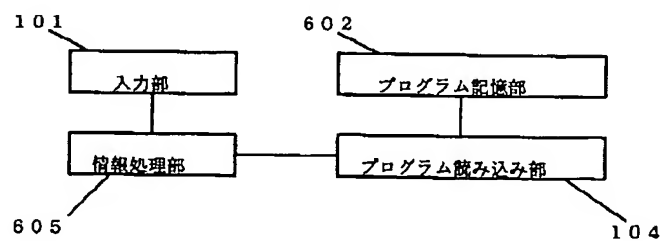
【図2】



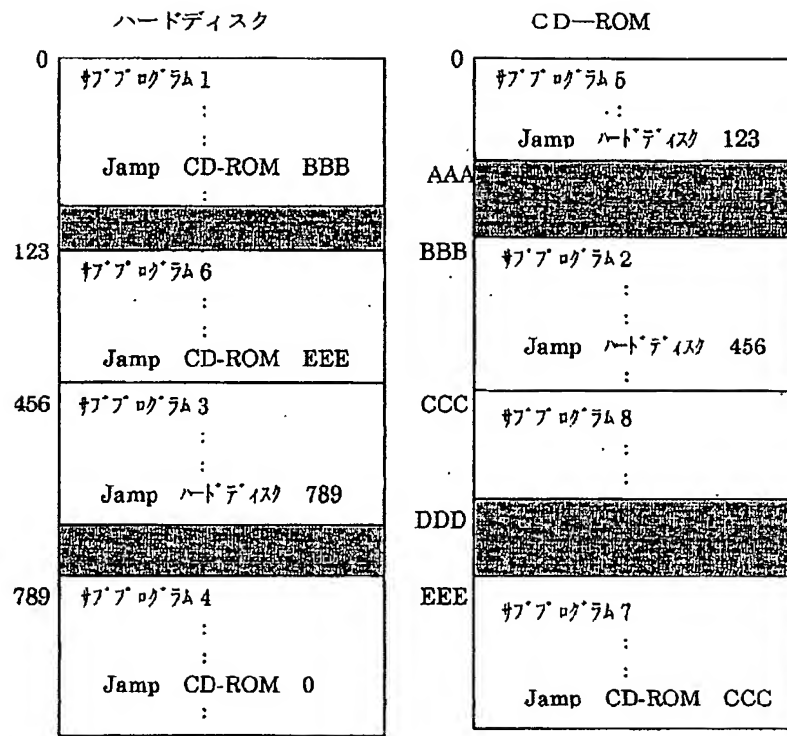
【図4】



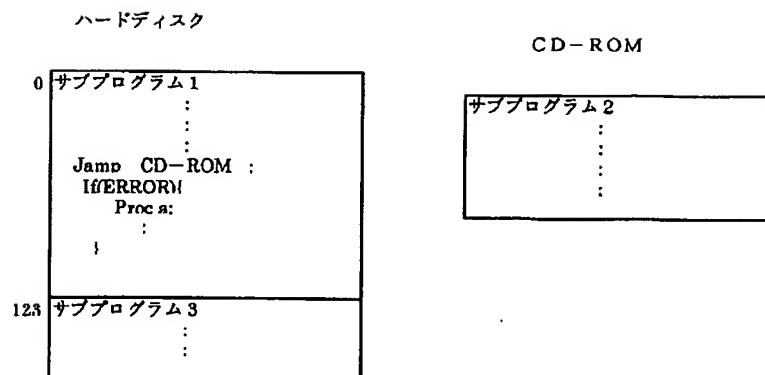
【図6】



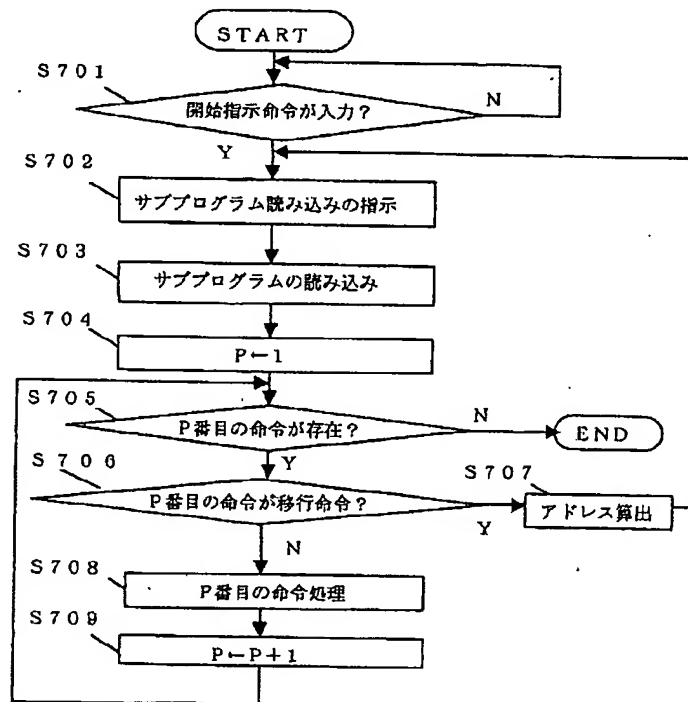
【図3】



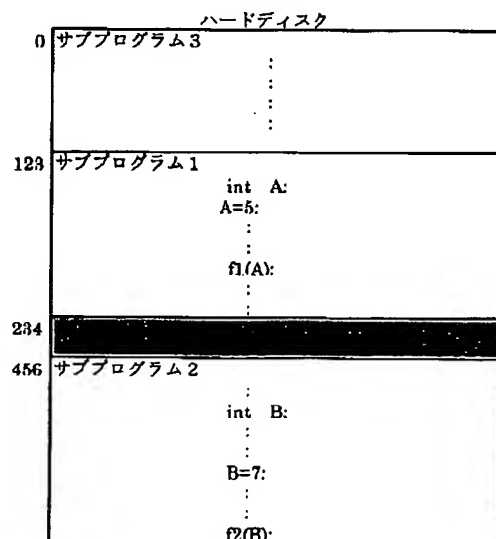
【図5】



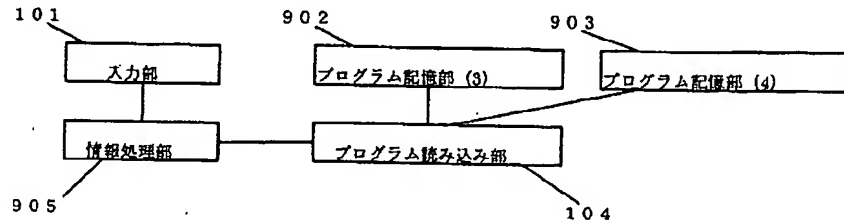
【図7】



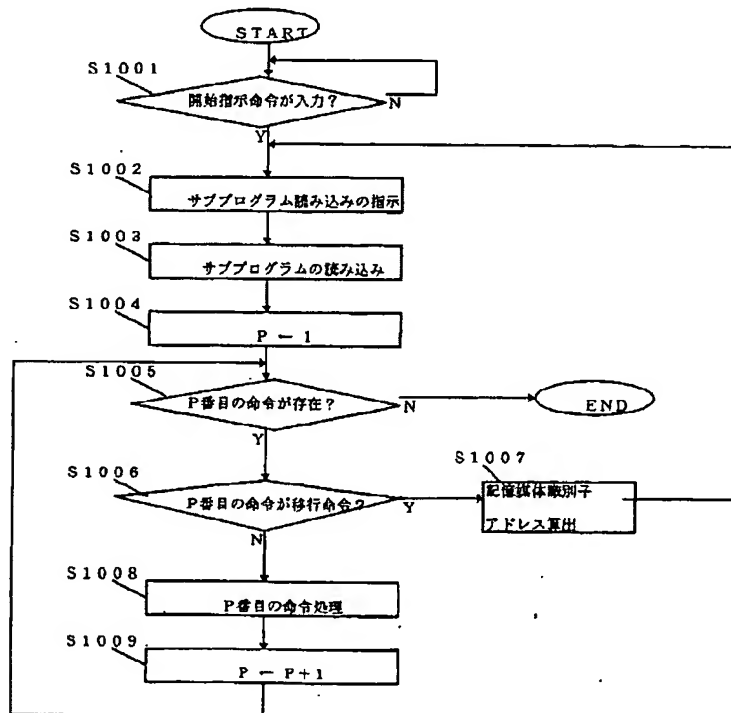
【図8】



【図9】



【図10】



【図11】

